2Ж

УДК 004.023

Т.А. Желдак, В.В. Слєсарєв
Державний ВНЗ «Національний гірничий університет», м. Дніпропетровськ, Україна
Україна, 49005, м. Дніпропетровськ, пр. Карла Маркса, 19

Алгоритм моделювання штучної імунної системи з селективним оператором Сааті та одновимірним локальним пошуком

T.A. Zheldak, V.V. Slesarev
State Higher Educational Institution «National Mining University», Dnipropetrovs'k, Ukraine
Ukraine, 49005, Dnipropetrovs'k, Karl Marks av., 19

The Algorithm of Artificial Immune System Simulation with Saaty Selection Operator and One-dimensional Local Search

Т.А. Желдак, В.В. Слесарев
Государственное ВУЗ «Национальный горный университет», г. Днепропетровск, Украина
Украина, 49005, г. Днепропетровск, пр. Карла Маркса, 19

Алгоритм моделирования искусственной иммунной системы с оператором селекции Саати и одномерным локальным поиском

Розглядається метод умовної оптимізації у безперервному просторі багатьох змінних, заснований на моделюванні імунної системи людини. Запропонований новий оператор селекції клітин для клонування, що базується на методі аналізу ієрархій (МАІ) Сааті. Для інтенсифікації роботи евристичного алгоритму використаний оператор одновимірного локального пошуку.
Ключові слова: імунні системи, оптимізація, Сааті, локальний пошук.

The method of constrained optimization in continuous space of several variables, based on the modeling of the human immune system is considered. New operator based on the Saaty Analytic Hierarchy Process (AHP) for cloning cells selection is proposed. Using one-dimensional local search operator to intensify the heuristic algorithm is offered.
Key words: immune systems, optimization, Saaty, local search.

Рассматривается метод условной оптимизации в непрерывном пространстве многих переменных, основанный на моделировании иммунной системы человека. Предложен новый оператор селекции клеток для клонирования, основанный на методе анализа иерархий (МАИ) Саати. Для интенсификации работы эвристического алгоритма использован оператор одномерного локального поиска.
Ключевые слова: иммунные системы, оптимизация, Саати, локальный поиск.

Вступ

Чимало інженерних, соціальних та економічних задач можуть бути сформульовані у вигляді задач оптимізації деяких цільових функцій багатьох змінних. При цьому в більшості випадків цільові функції виявляються мультимодальними. Наявність кількох

локальних оптимумів цільової функції одночасно з одним чи декількома глобальними оптимумами традиційно розглядається як суттєвий недолік. Існує цілий ряд методів і алгоритмічних прийомів, спрямованих на вихід з локальних оптимумів з метою досягнення єдиного глобального.

Задача ускладнюється, коли йдеться не про один, а про декілька критеріїв та множину рішень, недомінованих одне одним, серед яких все одно намагаються віднайти єдиний оптимум. В реальних застосуваннях часто існує необхідність пошуку не єдиного оптимального розв'язку задачі, а їх сімейства. Кожне з отриманих рішень, що в загальному випадку є субоптимальним, дозволяє розглянути одразу кілька можливих результатів.

Таке багатоваріантне розв'язання задачі дозволяє отримати метаевристика, відома як метод моделювання штучної імунної системи.

Запропонований нижче імунний алгоритм (ІА) імітує властивості природної імунної системи і заснований на принципах соматичної теорії [1] і мережевої гіпотези [2]. Соматична теорія стверджує, що збільшення різноманітності антитіл відбувається за рахунок соматичної рекомбінації і мутації генів. В рамках мережевої гіпотези обґрунтовується припущення, згідно з яким контроль розмноження клонів здійснюється в результаті взаємного розпізнавання антитіл, що функціонують як єдина мережа [1], [3], [4].

В процесі еволюції імунна система вищих ссавців набула здатності знищувати антигени за допомогою антитіл (рис. 1).

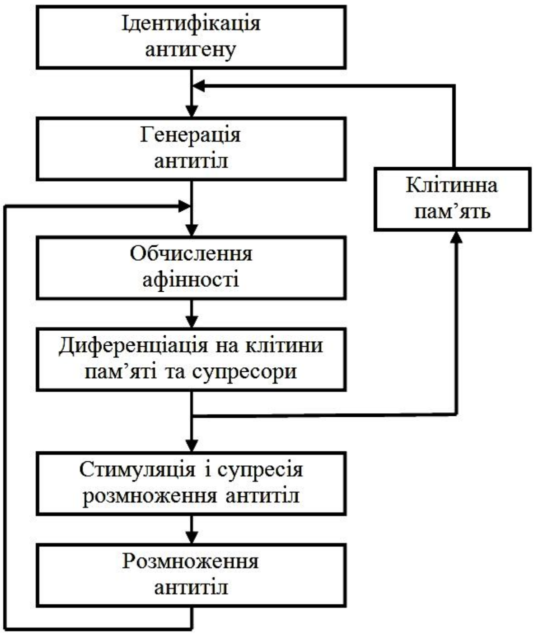


Рисунок 1 – Механізм формування антитіл в імунній системі

На рис. 1 показана робота імунної системи з точки зору процесів обробки і управління інформацією. Користуючись цією схемою, інформаційні процеси в імунній системі можна представити у вигляді алгоритму, який вкладається у наступну послідовність формалізованих кроків, докладно розглянутих у [3]:

1 Розпізнавання антигену. Цей процес відповідає встановленню вигляду задачі оптимізації $f^* = f(X^*) = \min_{X \in D \subseteq R^n} f(X)$, де n – розмірність вектора X дійсних змінних $x_i \in X$, заданих на області припустимих значень D.

2 Генерація антитіл клітинами пам'яті. Це відповідає пригадуванню успішного вирішення аналогічної (або схожої) задачі в минулому.

3 Обчислення афінності (корисності) клітини X_j, що є еквівалентним пошуку оптимального рішення: $\varphi_j = -f(X_j)$ для задачі мінімізації.

4 Диференціювання лімфоцитів: збереження відповідного рішення для наступного кроку пошуку та видалення зайвих кандидатів на рішення.

5 Стимулювання і придушення антитіл в залежності від афінності. Дослідження локальних оптимумів супроводжується підтриманням розмаїття напрямів пошуку.

6 Розмноження антитіл з використанням пам'яті та випадкових генів.

У природі описаний процес є безперервним – антитіла продукуються організмом як до відомого антигену, так і до його можливих мутацій. При моделюванні процесу на комп'ютері роботу алгоритму можна зупиняти після певної кількості поколінь або ж через заданий наперед час.

Результатом роботи алгоритму буде сімейство рішень, що відповідають антитілам у клітинах пам'яті, максимально пристосованих до антигену. У випадку пошуку екстремумів багатомодальних функцій це сімейство відповідатиме масиву з векторів-рішень, афінність яких максимально наближена до глобального оптимуму, тоді як самі рішення попарно відмінні не менш як на задану порогову величину.

Показниками ефективності роботи алгоритму можуть бути час виконання та кількість звернень до цільової функції при відшуканні глобального оптимуму, а також надійність (повторюваність) отриманих рішень при повторних запусках. Останнє є доволі важливим, адже евристичний метод, що розглядається, відноситься до класу методів випадкового пошуку і має збіжність лише за ймовірністю.

Метою даної роботи є розробка такого алгоритму, що реалізовував би метод моделювання штучної імунної системи для вирішення задачі багатовимірної умовної оптимізації багатомодальних безперервних функцій та забезпечував при цьому максимально можливі показники ефективності.

Аналіз існуючих алгоритмічних рішень

Історично першою реалізацією ідеї моделювання штучних імунних систем була ідея селекції та витіснення незалежних пошукових агентів, запропонована у алгоритмі CLONALG [5]. Даний алгоритм, що є поєднанням головних ідей генетичних алгоритмів та еволюційних стратегій, використовує двійкове кодування дійсного простору пошуку, селекцію певної частини (близько половини) кращих особин і їх рівномірне клонування, мутацію зворотно, пропорційну корисності, виключення найменш корисних антитіл та стиснення популяції.

Оператор стиснення популяції вилучає надлишковість популяції, виключаючи гірше з близьких рішень, що задовольняють вимозі

$$\|X_a - X_b\| < r, \quad a \neq b, \quad a,b \in [1:N_p], \tag{1}$$

де r – певний поріг близькості; $\|*\|$ – символ норми, яка розраховується в залежності від метрики простору; N_p – розмір популяції.

До недоліків алгоритму CLONALG слід віднести відсутність адаптації, обмежену точність, обумовлену двійковим кодуванням, та невикористання як такого механізму пам'яті.

Інший алгоритм моделювання імунних систем – алгоритм BCA [6] – фактично є версією еволюційної стратегії з двійковим кодуванням і оператором стиснення. Він використовує одразу два оператори мутації: у 3/4 випадків – бітову, а у 1/4 випадків – суміжну (кілька бітів, що стоять поруч, інвертуються одночасно). Метод показує високу

ефективність пошуку при невеликих розмірах популяції $N_p = 3 \div 5$, однак, як і CLONALG, на багатовимірних тестових задачах за якістю рішень програє відомим еволюційним стратегіям.

Певним кроком вперед є алгоритм opt-AiNet, головною особливістю якого є робота безпосередньо у дійсному просторі та збереження всіх знайдених локальних і глобальних оптимумів цільової функції у пам’яті, розмір якої динамічно змінюється. Алгоритм використовує також динамічний розмір популяції, керований механізмом стискання [3]. Здійснюється мутація кожного клону, пропорційно корисності його батьківської клітини за формулою

$$X_j^{C_i} = X_j^{C_i} + \alpha_j \cdot N(0;1), \quad \alpha_j = \rho e^{-\hat{\varphi}_j}, \quad j = [1 : N_p], \tag{2}$$

де $X_j^{C_i}$ – i-й клон j-ї клітини попередньої популяції; α_j – радіус мутації для клонів j-ї клітини; $\hat{\varphi}_j$ – відносна афінність j-ї клітини у поколінні $\hat{\varphi}_j = \frac{\varphi_{\max} - \varphi_j}{\varphi_{\max} - \varphi_{\min}}$; ρ – вільний позитивний параметр методу; $N(0;1)$ – нормально розподілена випадкова величина з математичним очікуванням 0 і середньоквадратичним відхиленням (СКВ), що дорівнює 1.

Практична реалізація даного алгоритму на ряді функцій [7] показала його значну перевагу над алгоритмом CLONALG у швидкості отримання порівняного за якістю рішення.

В роботі [4] запропоновано доповнити алгоритм opt-AiNet оператором випадкового локального пошуку. Останнє робить деякі реалізації алгоритму порівняними за ефективністю роботи з відомими генетичними алгоритмами та еволюційними стратегіями на класі розв’язуваних задач. Не випадково – і у цьому головна цінність opt-AiNet – він є основою для кількох більш сучасних алгоритмів, що реалізують викладений вище метод моделювання імунних систем.

При вирішенні багатьох прикладних задач з використанням методу моделювання імунних систем застосовується алгоритм HIA, запропонований в [1] і удосконалений в [7]. Його можна розглядати як модифікацію методу opt-AiNet. На відміну від останнього, розмір популяції в алгоритмі HIA залишається постійним в процесі всього ітераційного процесу й підтримується оператором стискання (1).

Крім того, кожна з клітин має додатковий атрибут t, званий віком клітини. За його допомогою визначаються так звані «елітні» клітини, що прожили $t \geq t_{\max}$.

Це – рішення, які тривалий час не були покращенні мутацією, тому вважаються локальними оптимумами. Вони заносяться в пам’ять й використовуються для генерації клонів у кожному новому поколінні.

Мутацію клонів здійснюють двома способами. Перший, відповідальний за інтенсифікацію пошуку в околі знайденого рішення, полягає у покоординатній зміні клона з використанням адаптивного нормального розподілу рівня мутації

$$X_j^{C_i} = X_j^{C_i} + N(0;\alpha_j), \tag{3}$$

де $\alpha_j(t)$ – СКВ нормального розподілу, що для кожної ітерації t визначається залежністю

$$\alpha_j(t) = \begin{cases} 2\left(X_j(t-1) - X_j^{C_i}(t-1)\right), & if\ f\left(X_j^{C_i}(t-1)\right) < f\left(X_j(t-1)\right); \\ \alpha_j(t-1) & overwise \end{cases}. \tag{4}$$

Другий спосіб має на меті диверсифікувати напрямок пошуку за рахунок обстеження ще не розглянутих ділянок і полягає у зміні однієї координати i випадковим числом з усієї області пошуку

$$x_{j,i} = U\left(x_{\min,i}; x_{\max,i}\right), \quad i = [1:n]. \tag{5}$$

Тут n – розмірність простору; $U\left(x_{\min,i}; x_{\max,i}\right)$ – рівномірно розподілена випадкова величина в межах $D = \left\{x_i \mid x_{i,\min} \le x_i \le x_{i,\max}\right\}$.

Алгоритм HIA, у порівнянні з попередніми, демонструє надійність пошуку глобального оптимуму багатоекстремальних функцій за доволі обмежений час.

Деякі дослідники, зокрема [8], [9], включають до алгоритмів моделювання імунних систем оператори локального пошуку, які базуються на наведених вище методах безумовної оптимізації. Втім, застосування методів сходження на гору, Нелдера-Міда та множників Лагранжа, зокрема, у методі SIA [8] для розв’язання задач з обмеженнями, хоч й посилює локальну збіжність алгоритмів, втім, не істотно впливає на ймовірність отримання глобального оптимуму.

У алгоритмі Dopt-AiNet [9] застосовано оператор копіювання координат, який працює наступним чином: якщо після мутації по якійсь із координат афінність клону покращилася, нове значення за цією координатою намагаються застосувати до решти вимірів. Такий оператор показує високу ефективність в разі симетрії функції по координатах, властивої для більшості тестових функцій, наприклад, Растрігіна, Еклі, Швефеля, Гріванка та інших [2]. Однак, при розв’язанні реальних задач, таких як навчання нейронних мереж прямого розповсюдження, чи основаних на радіально-базисних функціях, цей оператор абсолютно не ефективний.

Більш доцільним бачиться введення у загальний алгоритм роботи штучної імунної системи оператора інформаційного обміну між пошуковими агентами (рекомбінації). Останній, відомий нам з генетичного алгоритму під назвою кросоверу, дозволяє значно покращити якість рішень і досягати потрібної якості в реальному масштабі часу за рахунок інформаційного обміну.

Зокрема, у [10] пропонується для кожного з утворених клонів з невеликою ймовірністю $P_{cross} \le 0{,}2$ виконувати його кросовер з випадковою клітиною пам’яті. При цьому для кожного з генів рівноймовірно, залишаться вони без змін, чи будуть замінені на гени клітини пам’яті.

Удосконалений алгоритм оптимізації HINO-SF

Враховуючи розглянуті вище реалізації алгоритмів, котрі реалізують метод моделювання штучних імунних систем, пропонується наступний алгоритм, що не потребує двійкового кодування дійсних змінних та застосовує ряд операторів, відомих, зокрема, з генетичних та меметичних алгоритмів. Метою останніх має стати як інтенсифікація адаптивного локального пошуку, так і якомога повніше вивчення всієї області пошуку.

Назвемо запропонований алгоритм HINO-SF (*Hybrid Immune Network Optimization algorithm with Saaty selection and Fibonacci search*). Він складається з наступних кроків:

1 Випадково генерується популяція антитіл. Лічильник поколінь встановлюється в $t = 0$.

2 Якщо досягнуто наперед заданий максимум кількості поколінь, здійснюється перехід до кроку 11; в іншому випадку – перехід до кроку 3.

3 За оцінкою пристосованості клітин поточного покоління призначається кількість клонів для кожної з клітин поточного покоління – оператор селекції.

4 Клонування клітин у кількості, що визначена оператором селекції.

5 Рекомбінація клонованих особин за допомогою оператору ймовірнісного кросоверу.

6 Розрахунок динамічної ймовірності мутації для клонів та виконання адаптивного оператора мутацій.

7 З певною ймовірністю виконання для кожного клона оператору одновимірного локального пошуку за випадковою координатою методом золотого перетину (Фібоначчі).

8 Стиснення спільної популяції батьків та клонів за рахунок відповідного оператору до заданого рівня N_p.

9 Знищення з поточного покоління клітин, вік яких досяг заданого значення $t_{\max}$. Їх місце в основній популяції займають клітини, згенеровані випадково рівномірно в області пошуку.

10 Лічильник поколінь $t = t + 1$. Перехід на крок 2.

11 Виведення поточного покоління, як множини субоптимальних рішень.

12 Зупинка алгоритму.

Розглянемо кожен зі згаданих в алгоритмі операторів докладніше.

Оператор селекції. Для визначення кількості клонів, що припадає на кожну з особин поточного покоління, пропонується використовувати оператор, що застосовує метод аналізу ієрархій (МАІ) Сааті з відтинанням, вперше застосований у операторах клональної селекції еволюційних оптимізаційних стратегій [11]. Автор називає свій оператор h – зрізом множини клітин поточного покоління, що описується рівнянням

$$M_h = \{X_j \in D \mid \mu(f(X_j)) > h_{\min}\}, \quad j = \lfloor 1 : N_p \rfloor, \tag{6}$$

де $h_{\min} = 1/N_p$ - к – критична пристосованість клітини; D – область пошуку; $\mu(f(X_j))$ – функція належності клітини множині якісних.

Функція належності клітини визначається наступним чином:

– рішення нормуються за значенням афінності на відрізок $[0;1]$;

– розраховується рейтинг кожного з них $a_j = 1 + \lfloor \varphi(X_j) * k \rfloor$, де k – розмірність шкали (за умовчанням $k = 9$);

– будується матриця попарних порівнянь, для першого рядку якої застосовується вираз $b_{1i} = a_1 / a_i$, а для решти – $b_{ij} = b_{1j} / b_{1i}$;

– функції належності розраховуються як $\mu(f(X_i)) = 1 / \sum_j b_{ji}$;

– клітини, з $\mu(f(X_j)) > h_{\min}$, клонуються у кількості, пропорційній функції належності $n_c = N_c \cdot \mu(f(X_i))$, решта батьківського покоління відкидається.

Для ефективної роботи подальших операторів кросовера та мутації, бажано, аби особини з високою афінністю продукували достатньо велику кількість клонів. Тому рекомендується обирати $N_c = 10...20 \cdot N_p$.

Клональний відбір за МАІ є однією з ключових відмінностей запропонованого алгоритму від решти відомих реалізацій, у яких застосовується турнірний, пропорційний («рулетковий») чи рівномірний принципи розподілу кількості клонів між батьківськими особинами. Крім того, що даний метод, на відміну від решти, є математично обґрунтованим, його практичне застосування забезпечує кращу збіжність при вирішенні типових задач у порівнянні з іншими операторами селекції.

Оператор кросоверу. Іншою ключовою відмінністю запропонованого алгоритму є застосування до клонів адаптивного оператору кросоверу, що передбачає обмін генетичною інформацією. Оскільки було запропоновано не кодувати дійсну інформацію

двійковою, то для рекомбінації генів пропонується застосування модельованого двійкового кросоверу (Simulated binary crossover, SBX) [10], що дозволяє імунним клітинам виконувати склеювання векторів дійсних координат аналогічно двійковим векторам в генетичних алгоритмах.

Оператор SBX передбачає три параметри, що задаються користувачем:

– P_{cross} – ймовірність, що для клона буде виконано оператор кросоверу (зазвичай має значення, близьке до 0,5);

– P_{ind} – ймовірність того, що по даній координаті виконуватиметься схрещування (може варіюватися в широких межах від 0,4 до 1 – в останньому випадку кросовер виконується по всіх координатах одночасно);

– індекс варіації η – показує, як сильно нащадок схрещування має бути схожий на окремого батька (значення 0 та 1 з великою ймовірністю дадуть нащадка, схожого на одного з батьків, значення 0,5 – максимальне змішування генів).

Якщо випадкове число, рівномірно розподілене між 0 та 1 (тут і надалі позначаємо $U(0;1)$) менше ймовірності P_{cross} для двох випадкових клонів з номерами j та k, а інше випадкове число $U(0;1)$ менше P_{ind}, то поточна координата i клонів піддається схрещуванню за формулою

$$\begin{cases} x'_{j,i} = \frac{1}{2}\left((1+\beta)x_{j,i} + (1-\beta)x_{k,i}\right), \\ x'_{k,i} = \frac{1}{2}\left((1+\beta)x_{k,i} + (1-\beta)x_{j,i}\right), \end{cases} \tag{7}$$

де $\beta -$ так званий ступінь схрещування, що обчислюється за формулою

$$\beta = \begin{cases} \left(\alpha \cdot U(0;1)\right)^{\frac{1}{1+\eta}}, & if\ U(0;1) \le \frac{1}{\alpha}; \\ \left(\frac{1}{2-\alpha \cdot U(0;1)}\right)^{\frac{1}{1+\eta}}, & if\ U(0;1) > \frac{1}{\alpha} \end{cases}. \tag{8}$$

Параметр α визначається через ступінь відмінності j-ї та k-ї клітин по координаті i, а саме $\alpha = 2 - \dfrac{1}{1 + 2\dfrac{\min\left\{x_{\max,i} - \max\left(x_{j,i};x_{k,i}\right); \min\left(x_{j,i};x_{k,i}\right) - x_{\min,i}\right\}}{\max\left(x_{j,i};x_{k,i}\right) - \min\left(x_{j,i};x_{k,i}\right)}}$.

Оператор мутації. Головним оператором, що відповідає за пошук рішення, в запропонованому алгоритмі, як і в більшості реалізацій методу штучних імунних систем, залишається оператор мутації. Пропонується адаптивний оператор, який випадковим чином застосовує наступну мутацію

$$x^C_{j,i} = x_{j,i} + \delta_i \cdot \left(x_{\max,i} - x_{\min,i}\right), \quad i = [1:n], \tag{9}$$

де випадкова складова δ_i може визначатися на основі Гаусового розподілу згідно з (2) за формулою

$$\delta_i = 0{,}1 \cdot N(0;1), \tag{10}$$

або поліноміального розподілу за формулою

$$\delta_i = \begin{cases} \left(2 \cdot r_i + (1-2r_i)\Delta_i^{\mu+1}\right)^{\frac{1}{\mu+1}} - 1, & if \quad r_i < \frac{1}{2}; \\ 1 - \left(2 \cdot (1-r_i) + 2(r_i - 0{,}5)\Delta_i^{\mu+1}\right)^{\frac{1}{\mu+1}} - 1, & if \quad r_i \ge \frac{1}{2} \end{cases}, \tag{11}$$

де $r_i = U(0;1)$ – випадкове число, рівномірно розподілене для кожної з координат; $\Delta_i = \dfrac{\max\left\{x_{\max,i} - x_{j,i}; x_{j,i} - x_{\min,i}\right\}}{x_{\max,i} - x_{\min,i}}$ – відносне положення поточної клітини в області

пошуку по i-й координаті; μ – показник щільності поліноміального розподілу (більші значення відповідають меншому розсіву мутації).

Отримуючи на вході традиційні параметри $P_{mut}^{\min}$ – мінімальний рівень мутації, mut_rate – відносне значення рівня мутації (частини генів, які попадуть під зміну оператором) та $T_{\max}$ – максимальну припустиму кількість поколінь, алгоритм на кожній ітерації розраховує ступінь мутації генів

$$mut_level = \begin{cases} P_{mut}^{\min}\left(1 + mut_rate - mut_rate\dfrac{2t}{T_{\max}}\right), & if \quad t < \dfrac{T_{\max}}{2}; \\ P_{mut}^{\min}, \quad if \quad t \geq \dfrac{T_{\max}}{2} \end{cases}, \qquad (12)$$

де t – номер поточного покоління (ітерації).

Якщо при переборі координат випадкове число менше mut_level, координата піддається тій чи іншій мутації, інакше – залишається без змін.

Рекомендується задавати $P_{mut}^{\min}$ – на рівні 0,2…0,5, а mut_rate – на рівні 0,5…1. Тоді незалежно від обраного часу роботи алгоритму на початкових ітераціях мутація буде виконуватися для всіх клонів, на кінцевих – менше, ніж для половини.

Вибір – яку саме мутацію виконувати: широку за (10) чи стислу за (11) – здійснюється на основі синтетичної оцінки поточної клітини в масштабі оцінок корисності на поточній ітерації по всій популяції. Остання визначається для кожної клітині індивідуально за формулою

$$mut_select_j = \frac{1}{2}\left(1 - \gamma\frac{t}{T_{\max}}\right)\frac{\varphi_{\max} - \varphi_j}{\varphi_{\max} - \varphi_{\min}}, \quad j = \left[1 : N_c\right], \qquad (13)$$

де $\gamma = 0{,}8 \div 0{,}95$ – максимальна припустима доля поліноміальних мутацій в загальній кількості.

Якщо при зверненні до клона випадкове рівномірне число виявиться меншим mut_select, то для генів клона виконуватиметься лише мутація (10), інакше – (11).

З аналізу (13) зрозуміло, що ширша мутація за Гаусом застосовуватиметься для клітин з низькою афінністю й частіше – на початкових кроках оптимізації. Кількість мутацій за Гаусом зменшуватиметься лінійно з кількістю ітерацій та пропорційно афінності клітини на користь поліноміальної мутації. Слід звернути увагу, що вибір типу мутації здійснюється для всієї клітини, а лише потім перевіряється, чи буде виконуватись ця мутація для кожної з координат, за (12).

Оператор локального пошуку. Аби інтенсифікувати локальний пошук рішення і зробити його ще більш цілеспрямованим, до клона, який пройшов кросовер і мутацію з ймовірністю $P_{ls} = 0{,}1...0{,}25$, застосовується оператор одновимірного локального пошуку методом золотого перетину (Фібоначчі), який складається з наступних операцій [12]:

– рівноймовірно обирається одна з координат $k = \left[1 : n\right]$;

– рівноймовірно обирається напрямок пошуку – до нижньої границі області пошуку $\min X_j^{C_i}$ чи верхньої $\max X_j^{C_i}$;

– здійснюється одновимірний локальний пошук у напрямку від поточного розташування клона у просторі $X_j^{C_i}$ до границі області пошуку.

Якщо пошук покращує афінність рішення, знайдене рішення зберігають.

Оператор стиснення. По закінченні усіх пошукових операторів клітини нового покоління оцінюються за якістю. Разом з попереднім поколінням вони утворюють конф-

ліктну множину, з якої, використовуючи принцип (1), відбирають N_p клітин з найкращою афінністю, що лежать не ближче, ніж у радіусі r одна від одної. Для цього виконуються наступні операції:

– сортування клонів за зменшенням афінності;

– циклічно від першої клітини до спустошення множини ще не розглянутих;

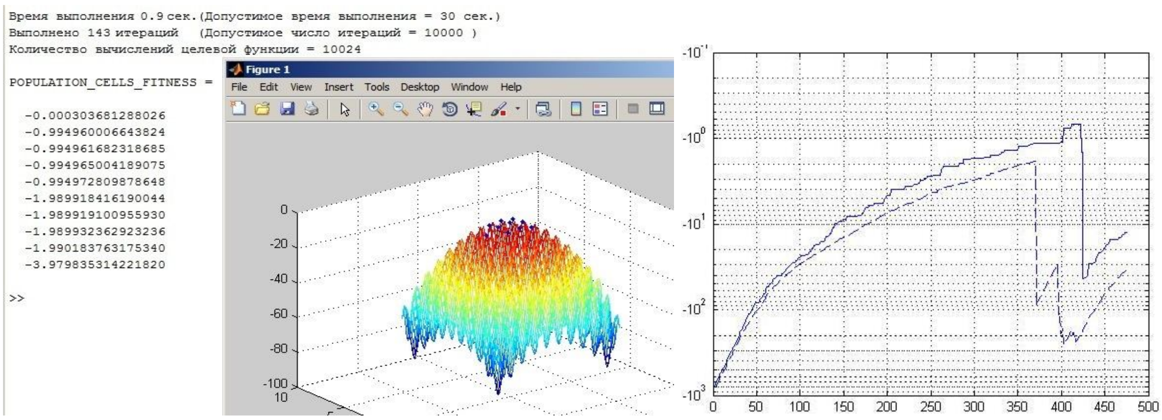
– вилучення усіх клітин, близьких до поточної і гірших за неї.

Слід наголосити також на необхідності вилучення після стиснення клітин, що мають вік, більший за визначену межу $t_{\max}$. Пропонується знищувати клітину з $t_{\max} = 10..20$ ітерацій, незалежно від її афінності. Подібні клітини замінюються новими, у яких кожна клітина генерується за (5).

Аналіз результатів роботи алгоритму

Для порівняння роботи запропонованого алгоритму з відомими реалізаціями методу моделювання штучних імунних систем та з іншими евристичними методами, що традиційно використовуються для вирішення задач багатовимірної оптимізації, було проведено тестування на ряді стандартних багатоекстремальних функцій. Також було порівняно роботу запропонованого алгоритму із відомими раніше при вирішенні практичної задачі – навчання нейронної мережі прямого розповсюдження.

На рис. 2а представлений знімок екрана результатів виконання алгоритму при мінімізації функції Растрігіна $F(X) = 20 + x_1^2 + x_2^2 - 10\cos(2\pi x_1) - 10\cos(2\pi x_2)$. Для показовості на графік виведено не саму цільову функцію, а зворотну їй величину – афінність клітин. З рис. 2 легко можна побачити основну особливість методу моделювання імунних систем – отримання в результаті не одного рішення, а одразу певної множини, до якої входять як глобальний оптимум, так і найближчі за афінністю локальні оптимуми.



а) б)

Рисунок 2 – Тестування запропонованого алгоритму на функції Растрігіна: (а) результат мінімізації $n = 2$; (б) зміна афінності клітин в часі $n = 20$

На рис. 2б показані залежності афінності кращої клітини (безперервна лінія) в поколінні (безперервна лінія) та середньої афінності популяції (пунктир) від номера ітерації при мінімізації функції Растрігіна 20 порядку. Графік, звичний для евристичних алгоритмів, має суттєву особливість: навіть найкращі рішення, якщо не були змінені протягом певного терміну тривалості життя, відкидаються й замінюються випадковими.

У табл. 1 наведені порівняльні результати роботи алгоритмів при розв’язанні типових задач. У якості задачі навчання нейронної мережі була використана задача ідентифікації механічних властивостей сортового прокату від його хімічного складу у скороченому та повному варіанті.

Таблиця 1 – Порівняння роботи алгоритмів при розв’язанні

Тестова функція (задача)	**Розмірність задачі (кількість змінних)**	**Метод розв’язання задачі**								
		RC-GA			**HIA**			**HINO-SF**		
		Показники якості реалізації алгоритму								
		f^*	$\tilde{f}^*$	$\tilde{n}^*$	f^*	$\tilde{f}^*$	$\tilde{n}^*$	f^*	$\tilde{f}^*$	$\tilde{n}^*$
Растрігіна	20	**0**	3,472	$4{,}4*10^6$	**0**	6,626	$2{,}4*10^5$	**0**	**1,929**	$\mathbf{1{,}0*10^5}$
	100	1,999	15,94	$1{,}6*10^7$	0,999	13,33	$8{,}6*10^6$	**0**	**2,616**	$\mathbf{2{,}0*10^6}$
Еклі	20	**0**	0,142	$2{,}0*10^6$	**0**	1,086	$7{,}2*10^5$	**0**	**0,007**	$\mathbf{8{,}1*10^4}$
	100	1,258	21,42	$2{,}1*10^7$	1,314	11,24	$9{,}4*10^6$	**1,101**	**7,240**	$\mathbf{2{,}1*10^6}$
Навчання нейромережі	72	4,974	16,37	$1{,}4*10^6$	2,994	11,33	$2{,}4*10^6$	**2,039**	**10,20**	$\mathbf{2{,}1*10^5}$
	144	14,94	39,52	$5{,}1*10^8$	13,99	21,07	$1{,}4*10^8$	**12,07**	**16,75**	$\mathbf{4{,}1*10^6}$

В табл. 1 прийняті позначення: f^* – мінімальне значення цільової функції, знайдене методом за 10 повторних запусків; $\tilde{f}^*$ – статистичне значення оцінки математичного очікування мінімального значення функції при випадковому запуску; $\tilde{n}^*$ – статистичне значення оцінки математичного очікування кількості викликів цільової функції до завершення алгоритму при випадковому запуску.

Як видно, запропонований алгоритм знаходить глобально оптимальні рішення частіше від аналогів, при цьому витрачаючи значно менше часу. На особливо складних задачах, як, наприклад, навчання нейронної мережі на реальних даних, виграш часу може становити майже два порядки.

Аби зрозуміти ефективність кожного з операторів, що складають алгоритм, зведемо у табл. 2 частоту, з якою вони покращують значення афінності клітин.

Таблиця 2 – Ефективність операторів запропонованого алгоритму

Оператор	Мутація Гауса	Поліноміальна мутація	Локальний пошук	Кросовер
Випадки покращення цільової функції, %	23,33%	47,99%	17,23%	11,45%

Слід відзначити, що отримані дані вповні відповідають теоретичним засадам, адже саме мутація є головним інструментом локального пошуку в методі моделювання штучних імунних систем. Відсутність у табл. 2 запропонованого оператора селекції пояснюється тим, що він не призводить до покращення рішень, а лише обирає кандидатів для клонування.

Висновки

Аналіз основних сучасних алгоритмів, що реалізують метод моделювання штучних імунних систем показав його переваги перед іншими методами багатовимірної умовної оптимізації у дійсному просторі. Запропоновано гібридний адаптивний імунний алгоритм, що використовує оператор клональної селекції, кросовер, адаптивну мутацію і обмежений локальний пошук.

Селекція здійснюється шляхом оцінки пристосованості рішень з використанням методу ієрархії Сааті. Запропонований оператор селекції ефективніший, ніж традиційні оператори, крім цього, він є єдиним математично обґрунтованим.

Парний адаптивний кросовер клітин-клонів забезпечує інтенсифікацію локального пошуку, дозволяючи поєднувати кращі гени двох вдалих рішень з попередніх поколінь. Аби підвищити спрямованість пошуку, до випадкової координати клона застосовується оператор одновимірної оптимізації методом золотого перетину (Фібоначчі).

Оператор покоординатної мутації застосовує нормальний або поліноміальний закони розподілу ймовірностей. При цьому, якщо клон породжений кліткою з високою афінністю, вищою є ймовірність «вузької» поліноміальної мутації, що відповідає інтенсивному локального пошуку. Клон, породжений клітиною зі слабкою афінністю, має більшу ймовірність «широкої» мутації з відхиленням за законом Гауса.

Результати дослідження показують високу ефективність запропонованого алгоритму для стандартних цільових функцій, що мають розмірність до 100, а також при вирішенні завдань навчання нейронних мереж прямого поширення.

Основні переваги запропонованого алгоритму полягають у тому, що він залишається ефективним при зростанні розмірності задачі, знаходить не одне рішення, а їх множину (альтернативи) та використовує значно менше часу (на порядок) для порівняного розв’язання задачі.

Література

1. Lucinska M. Hybrid Immune Algorithm for Multimodal Function Optimization : [текст] / M. Lucinska, S.T. Wierzchon // Recent Advances in Intelligent Information Systems. – 2009. – P. 301-313.
2. Rowan T.H. Functional Stability Analysis of Numerical Algorithms, Ph.D. Thesis, Department of Computer Sciences / T.H. Rowan [Електронний документ (ресурс)]. – University of Texas at Austin, 1990. – Режим доступу за URL : http://reference.kfupm.edu.sa/content/f/u/ functional_stability_analysis_of_numeric_1308737.pdf
3. De Castro L. An Artificial Immune Network for Multimodal Function Optimization : [текст] / L.N. De Castro, J. Timmis // Proceedings of IEEE Congress on Evolutionary Computation (CEC'02). – 2002. – Vol. 1. – P. 699-674.
4. Yildiz A.R. A novel hybrid immune algorithm for global optimization in design and manufacturing : [текст] // Robotics and Computer-Integrated Manufacturing. – 2009. – Vol. 25. – P. 261-270.
5. Brownlee J. Clonal Selection Algorithms : [Текст] / J. Brownlee // Technical Report 070209A, Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, 2007.
6. Kelsey J. Immune Inspired Somatic Contiguous Hypermutation for Function Optimisation / J. Kelsey, J. Timmis [Електронний документ (ресурс)]. – 2003. – Режим доступу за URL : http://www.cs.york.ac.uk/rts/docs/ GECCO_2003/papers/2723/ 27230207.pdf.
7. Chen J. A hybrid immune multiobjective optimization algorithm : [текст] / Jianyong Chen, Qiuzhen Lin, Zhen Ji // European Journal of Operational Research. – 2010. – Vol. 204. – P. 294-302.
8. Карпенко А.П. Гибридный метод глобальной оптимизации на основе искусственной иммунной системы : [текст] / А.П. Карпенко, Д.Л. Шуров // Инженерное образование. – 2012. – № 8.
9. de França F.O. An artificial immune network for multimodal function optimization on dynamic environments : [текст] / F.O. de França, F.J.V. Zuben, L.N. de Castro // In: Proc. of GECCO. – ACM Press, New York. – 2005. – P. 289-296.
10. Gong M. Multiobjective immune algorithm with nondominated neighbor-based selection / Maoguo Gong, Licheng Jiao, Haifeng Du, Liefeng Bo // Evolutionary Computation. – 2008. – Vol. 16(2). – P. 225-255.
11. Снитюк В.Є. Спрямована оптимізація і особливості еволюційної генерації потенційних розв’язків : [текст] / В.Є. Снитюк // VI міжнародна школа-семінар «Теорія прийняття рішень», Ужгород, 1 – 6 жовтня 2012 р. – Ужгород : «Інвізор», 2012 – С. 182-183.
12. Желдак Т.А. Метод моделювання штучної імунної системи в задачах оптимізації мультимодальних функцій : [текст] / Т.А. Желдак // Обчислювальний інтелект (результати, проблеми, перспективи) : Матеріали 2-ї міжнар. наук.-техн. конф. 14 – 17 травня 2013 р. – Черкаси : Маклаут, 2013. – С. 33-36.

Literatura

1. Lucinska M., Wierzchon S.T. Recent Advances in Intelligent Information Systems, 2009, pp. 301-313.

2. Rowan T.H. Ph.D. Thesis, Department of Computer Sciences, University of Texas at Austin, 1990. URL: http://reference.kfupm.edu.sa/content/f/u/functional_stability_analysis_of_numeric_1308737.pdf
3. De Castro L.N., Timmis J. Proceedings of IEEE Congress on Evolutionary Computation (CEC'02). 2002. – vol. 1. – P. 699-674.
4. Yildiz A.R. Robotics and Computer-Integrated Manufacturing. – 2009. – vol. 25. – P. 261-270.
5. Brownlee J. Technical Report 070209A, Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, 2007.
6. Kelsey J., Timmis J. Immune Inspired Somatic Contiguous Hypermutation for Function Optimisation. 2003. URL: http://www.cs.york.ac.uk/rts/docs/GECCO_2003/papers/2723/27230207.pdf .
7. Chen J., Q. Lin, Zh. Ji European Journal of Operational Research. 2010. Vol. 204. pp. 294-302.
8. Karpenko A.P., Shurov D.L. Inzhenernoe obrazovanie. – 2012. – Vol. 8.
9. F.O. de França, F.J.V. Zuben, L.N. de Castro In: Proc. of GECCO. ACM Press, New York. 2005. pp. 289-296.
10. Gong, M., Jiao, L., Du, H., & Bo, L. Evolutionary Computation. – 2008. – Vol. 16(2). – P. 225-255.
11. Snytyuk V.Y. VI Mizhnarodna shkola-seminar «Teorija pryjnjattja rishen». Uzhgorod, 1 – 6 zhovtnja 2012. – Uzhgorod, Invizor. – 2012. – s. 182-183.
12. Zheldak T.A. Obchysljuvalnyj Intelekt (rezultaty, problem, perspektyvy). Materialy 2d mizhnarodnoji konferenciji 14 – 17 travnja 2013 r. – Cherkasy : Maklaut, 2013. – s. 33-36.

RESUME

T.A. Zheldak, V.V. Slesarev

The Algorithm of Artificial Immune System Simulation with Saaty Selection Operator and One-dimensional Local Search

In the article the basic modern algorithm that implement the artificial immune systems simulation method is considered. Also advantages and disadvantages of this algorithms are analyzed.

Using a hybrid adaptive immune algorithm that includes the clonal selection, crossover, mutation and limited adaptive local search is proposed as a new algorithm.

Selection is carried out by estimating the solutions suitability using the Saaty Analytic Hierarchy Process (AHP). The higher is the affinity of the cells, which describes the solution, the more number of clones, the cell will provide the next generation.

The investigation shows that the proposed selection operator is more effective than the traditional operators, such as the tournament, roulette and uniform with cut-off. Furthermore, this selection method is only mathematically justified.

Twin adaptive cell clones crossover provides an intensification of local search and allows to combine the genes of two the best successful decisions of previous generations.

To improve the focus of the search the one-dimensional optimization operator based on the method of golden ratio (named Fibonacci) is applied to a random coordinate of clone cells.

The crossover and local search operators are applied to the clone cells with a certain probability, which can also change adaptively during the work of the algorithm.

Coordinatewise mutation operator with the normal and polynomial probability distributions applies. Moreover, if a clone is generated by a cell with high affinity, narrow polynomial mutation is more likely for it. It corresponds to intensive local search. If a clone is generated by a cell with a weak affinity, a broad mutation with the Gaussian distribution is more tredible.

The results show the high efficiency of the proposed algorithm for standard objective functions with less than 100 dimensions, as well as for solving the problem of feedforward neural networks training.

Стаття надійшла до редакції 27.06.2013.